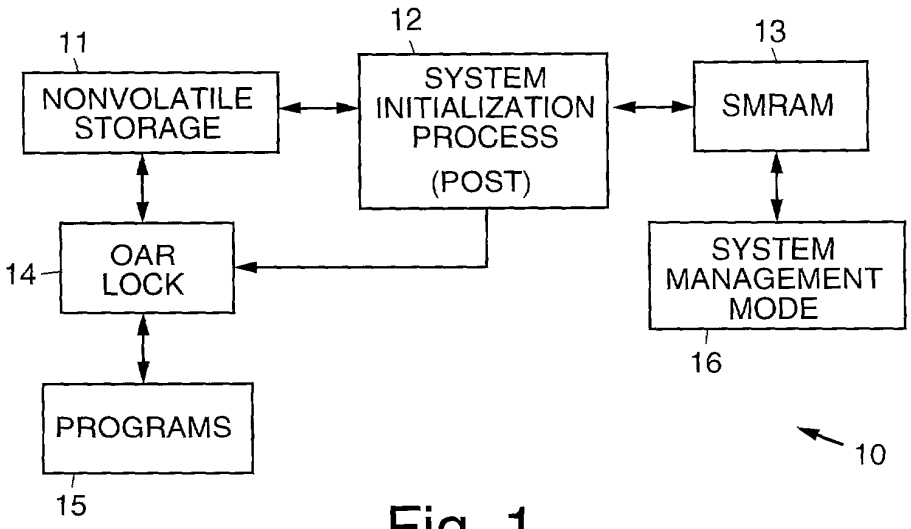

11
NONVOLATILE STORAGE
12
SYSTEM INITIALIZATION PROCESS (POST)
13
SMRAM
14
OAR LOCK
SYSTEM MANAGEMENT MODE
16
PROGRAMS
15
10

Fig. 1

Fig. 5

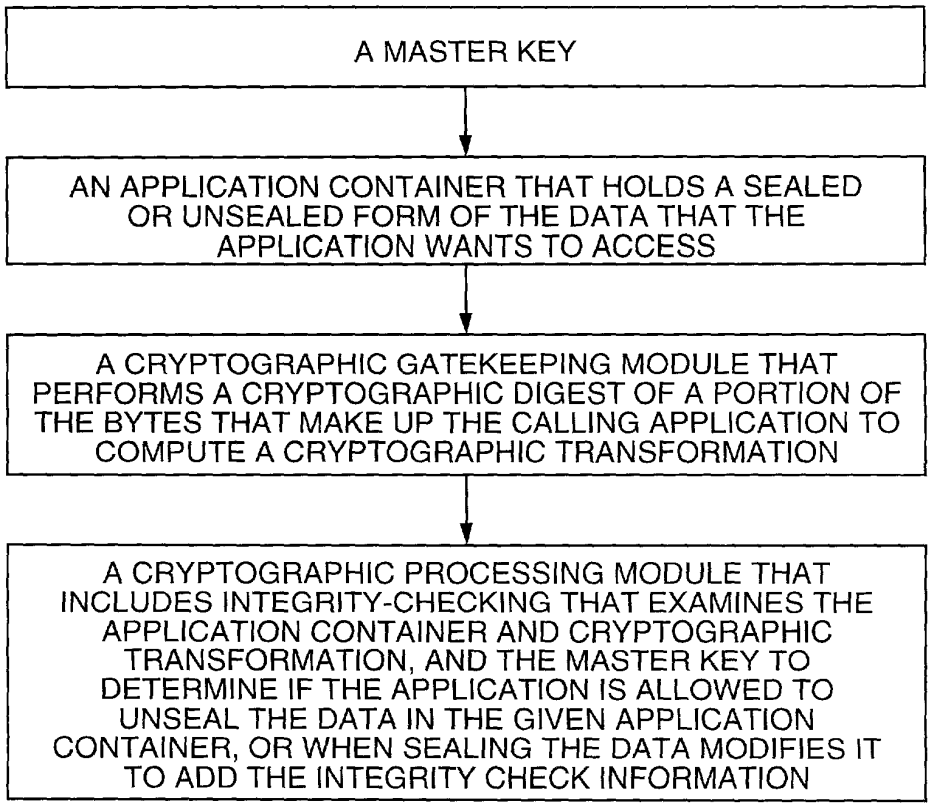

A MASTER KEY
AN APPLICATION CONTAINER THAT HOLDS A SEALED OR UNSEALED FORM OF THE DATA THAT THE APPLICATION WANTS TO ACCESS
A CRYPTOGRAPHIC GATEKEEPING MODULE THAT PERFORMS A CRYPTOGRAPHIC DIGEST OF A PORTION OF THE BYTES THAT MAKE UP THE CALLING APPLICATION TO COMPUTE A CRYPTOGRAPHIC TRANSFORMATION
A CRYPTOGRAPHIC PROCESSING MODULE THAT INCLUDES INTEGRITY-CHECKING THAT EXAMINES THE APPLICATION CONTAINER AND CRYPTOGRAPHIC TRANSFORMATION, AND THE MASTER KEY TO DETERMINE IF THE APPLICATION IS ALLOWED TO UNSEAL THE DATA IN THE GIVEN APPLICATION CONTAINER, OR WHEN SEALING THE DATA MODIFIES IT TO ADD THE INTEGRITY CHECK INFORMATION

Fig. 2

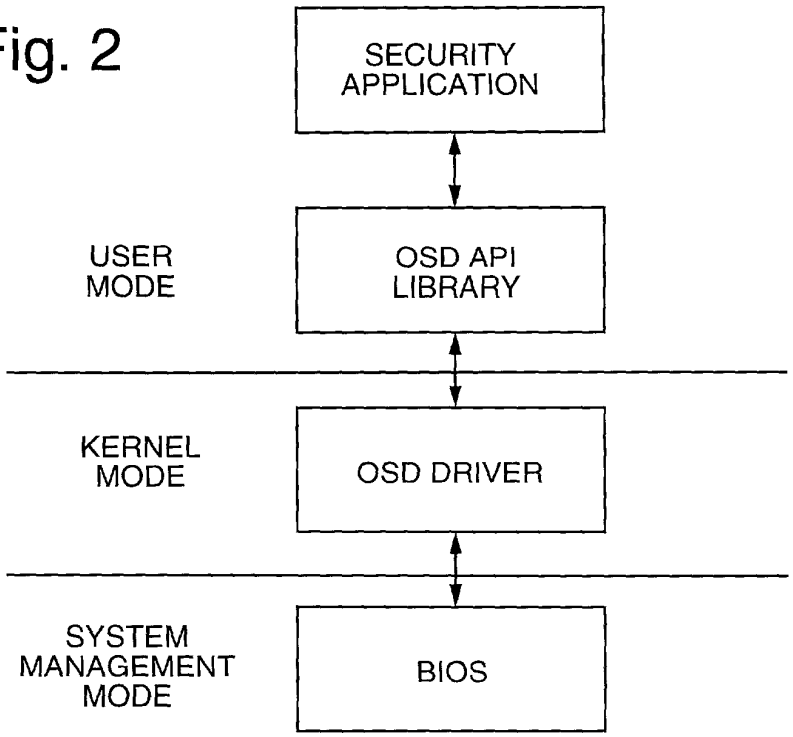
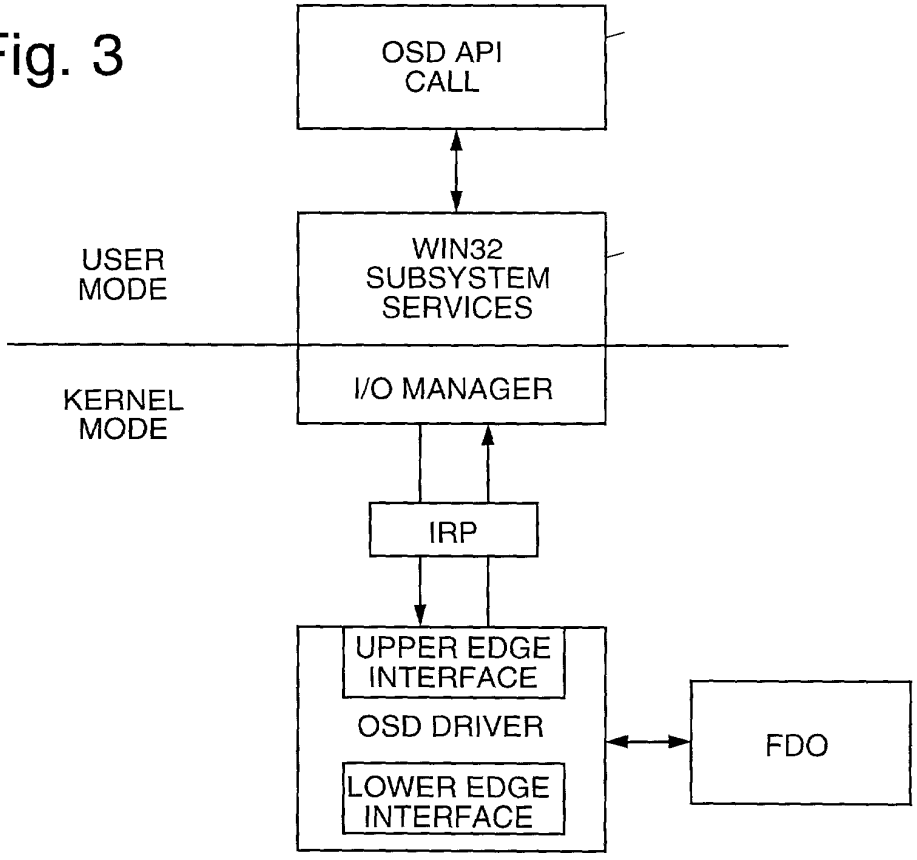

SECURITY APPLICATION
USER MODE
OSD API LIBRARY
KERNEL MODE
OSD DRIVER
SYSTEM MANAGEMENT MODE
BIOS

Fig. 3


OSD API CALL
USER MODE
WIN32 SUBSYSTEM SERVICES
KERNEL MODE
I/O MANAGER
IRP
UPPER EDGE INTERFACE
OSD DRIVER
LOWER EDGE INTERFACE
FDO

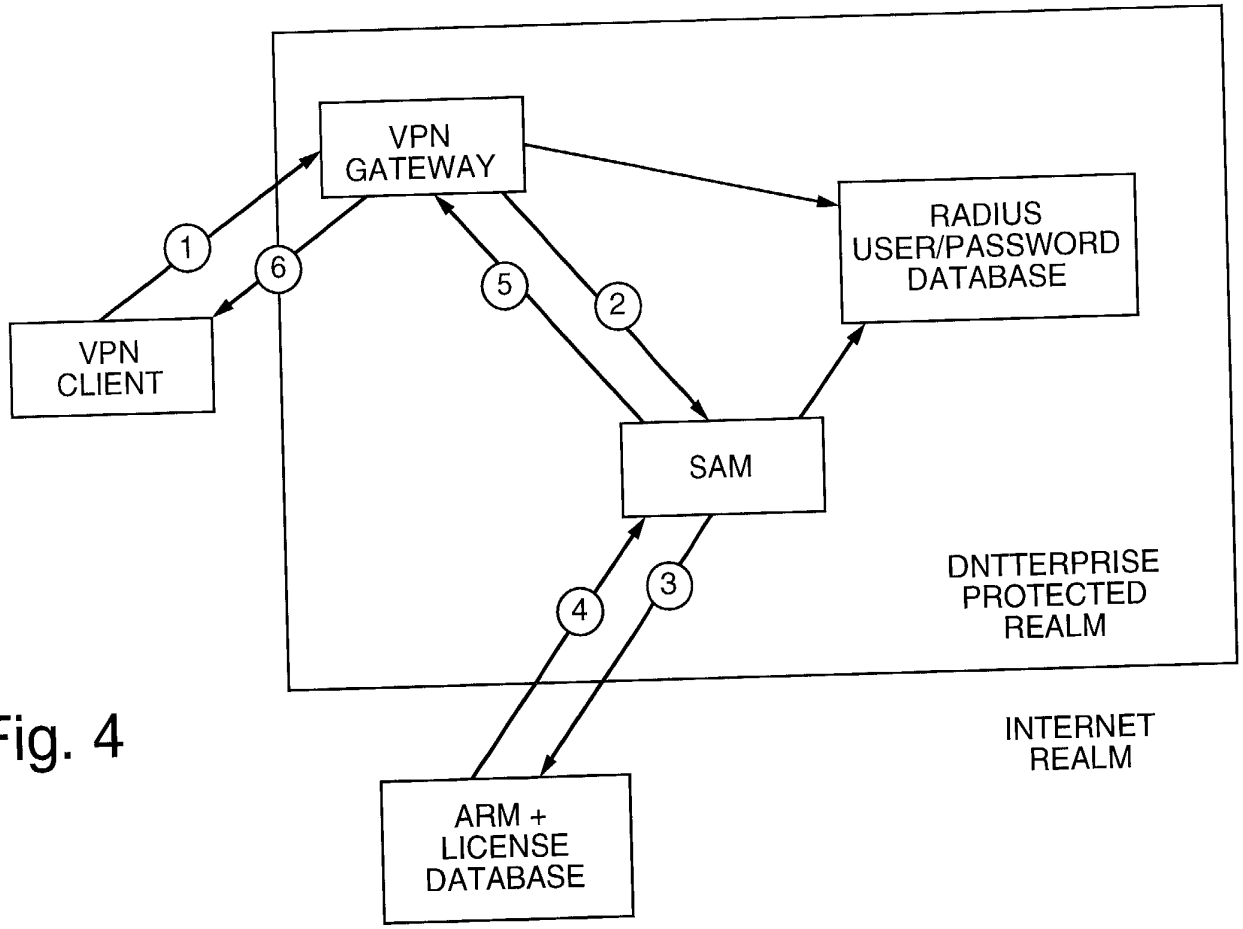

VPN
GATEWAY
RADIUS
USER/PASSWORD
DATABASE
VPN
CLIENT
SAM
DNTTERPRISE
PROTECTED
REALM
INTERNET
REALM
ARM +
LICENSE
DATABASE
1
2
3
4
5
6

Fig. 4

Fig. 6

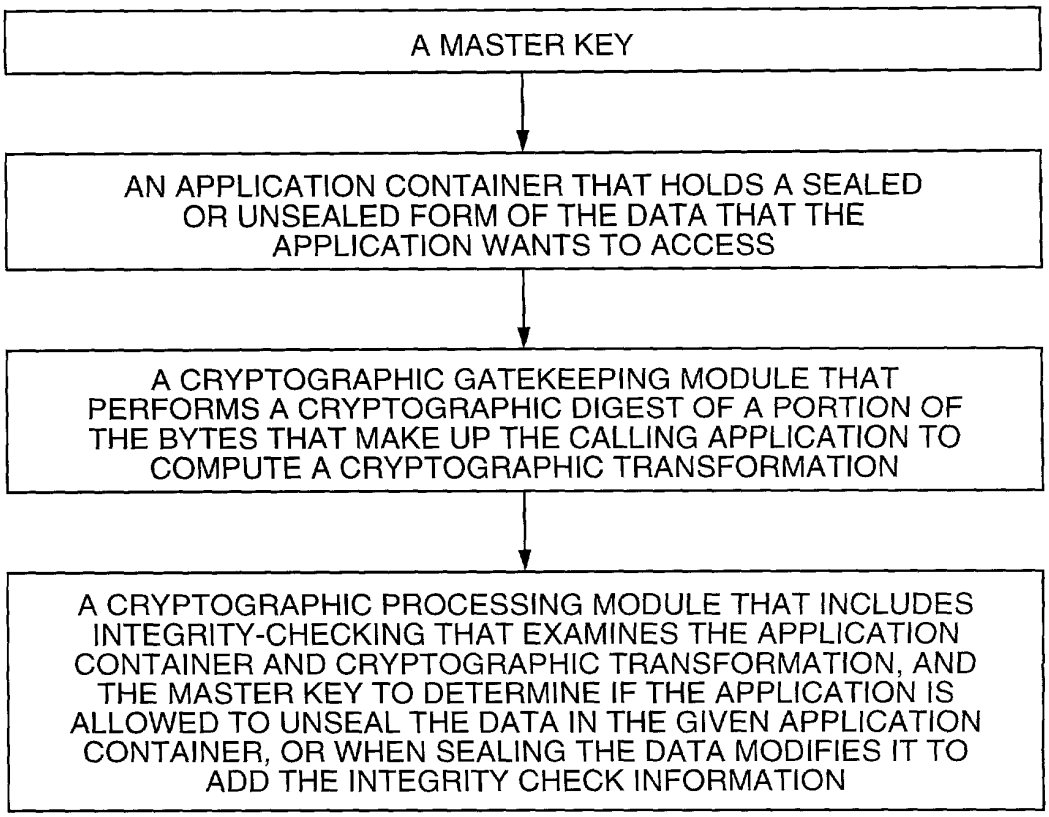

A MASTER KEY
AN APPLICATION CONTAINER THAT HOLDS A SEALED OR UNSEALED FORM OF THE DATA THAT THE APPLICATION WANTS TO ACCESS
A CRYPTOGRAPHIC GATEKEEPING MODULE THAT PERFORMS A CRYPTOGRAPHIC DIGEST OF A PORTION OF THE BYTES THAT MAKE UP THE CALLING APPLICATION TO COMPUTE A CRYPTOGRAPHIC TRANSFORMATION
A CRYPTOGRAPHIC PROCESSING MODULE THAT INCLUDES INTEGRITY-CHECKING THAT EXAMINES THE APPLICATION CONTAINER AND CRYPTOGRAPHIC TRANSFORMATION, AND THE MASTER KEY TO DETERMINE IF THE APPLICATION IS ALLOWED TO UNSEAL THE DATA IN THE GIVEN APPLICATION CONTAINER, OR WHEN SEALING THE DATA MODIFIES IT TO ADD THE INTEGRITY CHECK INFORMATION

Fig. 7

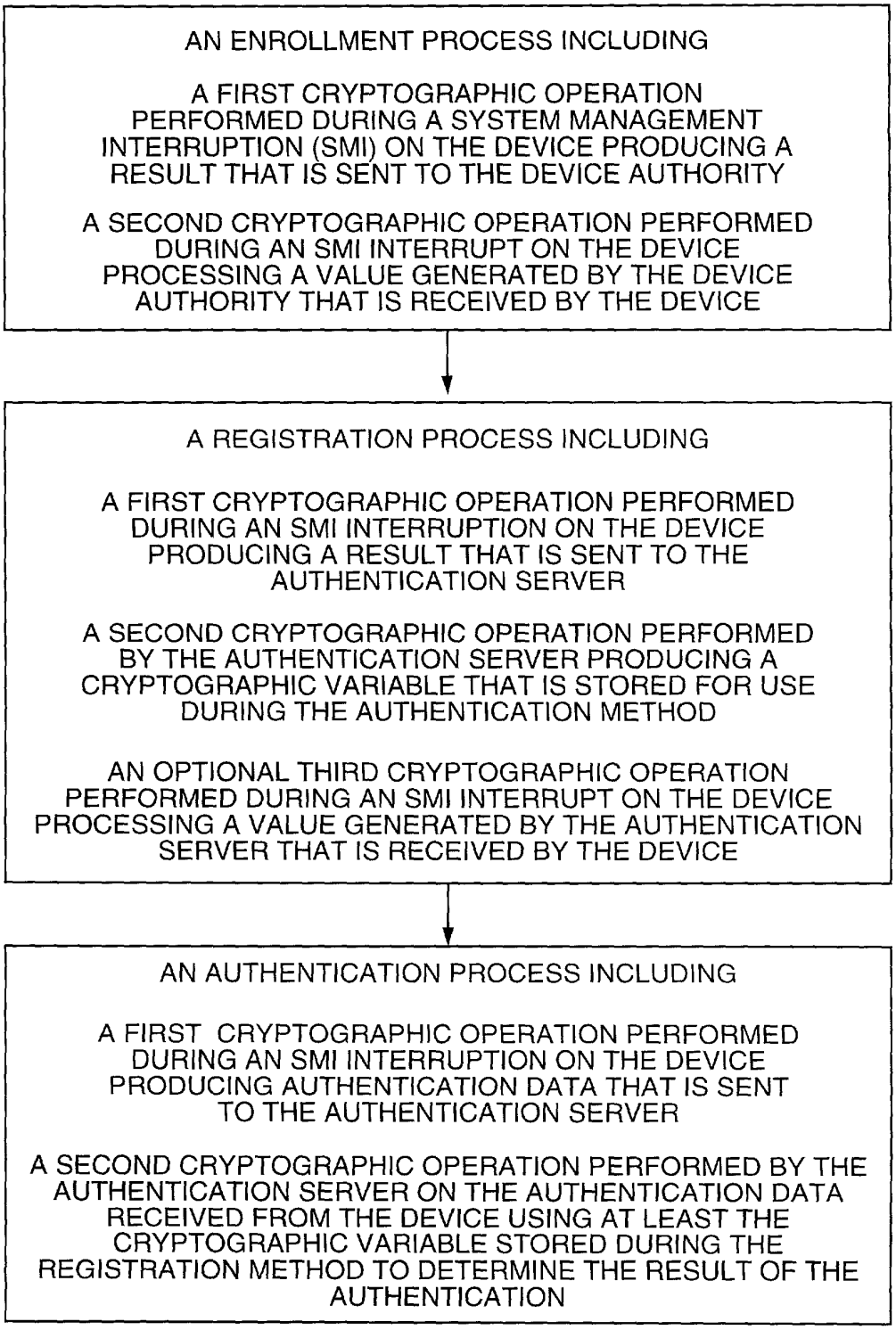

AN ENROLLMENT PROCESS INCLUDING
A FIRST CRYPTOGRAPHIC OPERATION PERFORMED DURING A SYSTEM MANAGEMENT INTERRUPTION (SMI) ON THE DEVICE PRODUCING A RESULT THAT IS SENT TO THE DEVICE AUTHORITY
A SECOND CRYPTOGRAPHIC OPERATION PERFORMED DURING AN SMI INTERRUPT ON THE DEVICE PROCESSING A VALUE GENERATED BY THE DEVICE AUTHORITY THAT IS RECEIVED BY THE DEVICE
A REGISTRATION PROCESS INCLUDING
A FIRST CRYPTOGRAPHIC OPERATION PERFORMED DURING AN SMI INTERRUPTION ON THE DEVICE PRODUCING A RESULT THAT IS SENT TO THE AUTHENTICATION SERVER
A SECOND CRYPTOGRAPHIC OPERATION PERFORMED BY THE AUTHENTICATION SERVER PRODUCING A CRYPTOGRAPHIC VARIABLE THAT IS STORED FOR USE DURING THE AUTHENTICATION METHOD
AN OPTIONAL THIRD CRYPTOGRAPHIC OPERATION PERFORMED DURING AN SMI INTERRUPT ON THE DEVICE PROCESSING A VALUE GENERATED BY THE AUTHENTICATION SERVER THAT IS RECEIVED BY THE DEVICE
AN AUTHENTICATION PROCESS INCLUDING
A FIRST CRYPTOGRAPHIC OPERATION PERFORMED DURING AN SMI INTERRUPTION ON THE DEVICE PRODUCING AUTHENTICATION DATA THAT IS SENT TO THE AUTHENTICATION SERVER
A SECOND CRYPTOGRAPHIC OPERATION PERFORMED BY THE AUTHENTICATION SERVER ON THE AUTHENTICATION DATA RECEIVED FROM THE DEVICE USING AT LEAST THE CRYPTOGRAPHIC VARIABLE STORED DURING THE REGISTRATION METHOD TO DETERMINE THE RESULT OF THE AUTHENTICATION

Fig. 8

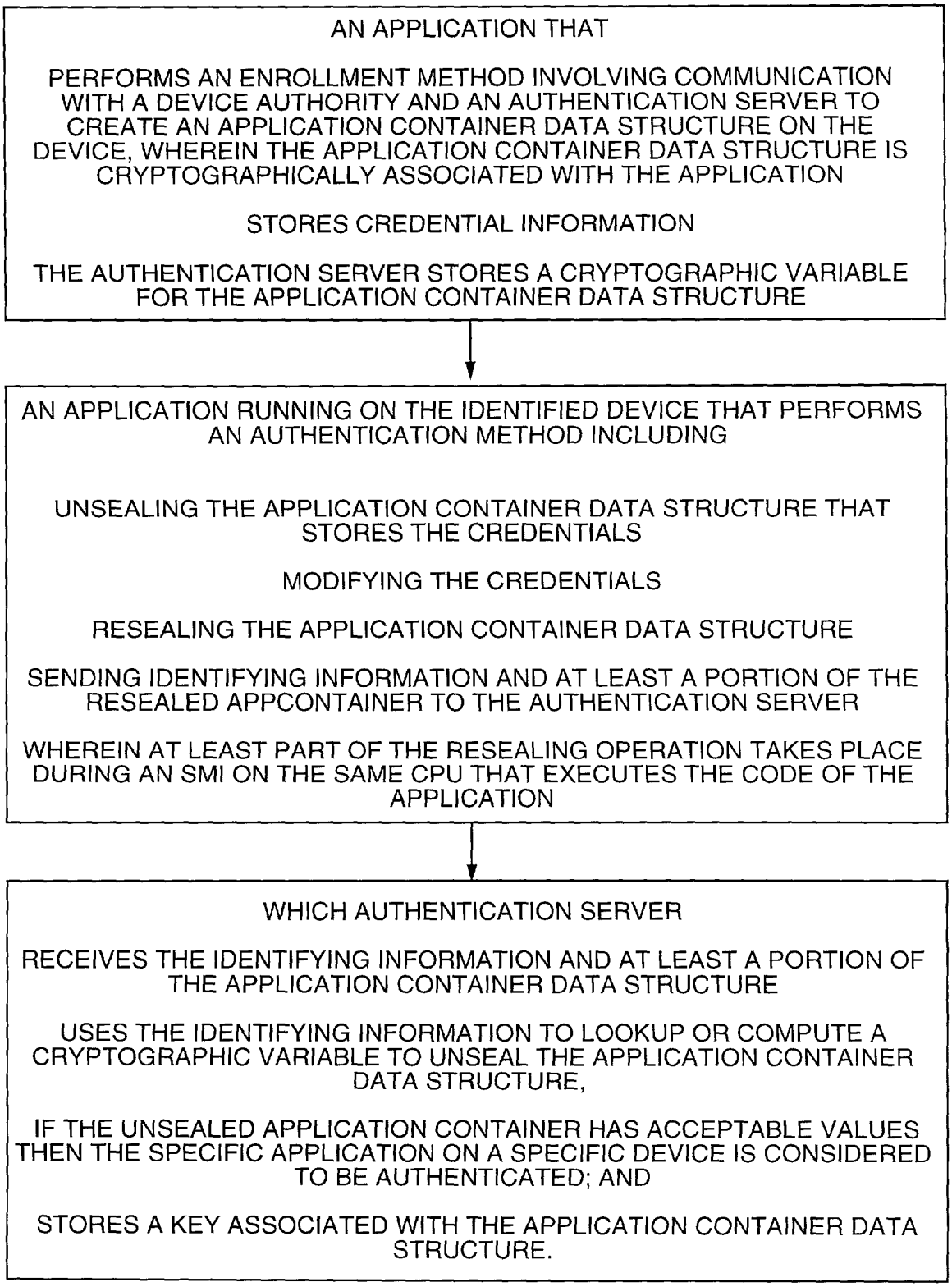

AN APPLICATION THAT
PERFORMS AN ENROLLMENT METHOD INVOLVING COMMUNICATION WITH A DEVICE AUTHORITY AND AN AUTHENTICATION SERVER TO CREATE AN APPLICATION CONTAINER DATA STRUCTURE ON THE DEVICE, WHEREIN THE APPLICATION CONTAINER DATA STRUCTURE IS CRYPTOGRAPHICALLY ASSOCIATED WITH THE APPLICATION
STORES CREDENTIAL INFORMATION
THE AUTHENTICATION SERVER STORES A CRYPTOGRAPHIC VARIABLE FOR THE APPLICATION CONTAINER DATA STRUCTURE
AN APPLICATION RUNNING ON THE IDENTIFIED DEVICE THAT PERFORMS AN AUTHENTICATION METHOD INCLUDING
UNSEALING THE APPLICATION CONTAINER DATA STRUCTURE THAT STORES THE CREDENTIALS
MODIFYING THE CREDENTIALS
RESEALING THE APPLICATION CONTAINER DATA STRUCTURE
SENDING IDENTIFYING INFORMATION AND AT LEAST A PORTION OF THE RESEALED APPCONTAINER TO THE AUTHENTICATION SERVER
WHEREIN AT LEAST PART OF THE RESEALING OPERATION TAKES PLACE DURING AN SMI ON THE SAME CPU THAT EXECUTES THE CODE OF THE APPLICATION
WHICH AUTHENTICATION SERVER
RECEIVES THE IDENTIFYING INFORMATION AND AT LEAST A PORTION OF THE APPLICATION CONTAINER DATA STRUCTURE
USES THE IDENTIFYING INFORMATION TO LOOKUP OR COMPUTE A CRYPTOGRAPHIC VARIABLE TO UNSEAL THE APPLICATION CONTAINER DATA STRUCTURE,
IF THE UNSEALED APPLICATION CONTAINER HAS ACCEPTABLE VALUES THEN THE SPECIFIC APPLICATION ON A SPECIFIC DEVICE IS CONSIDERED TO BE AUTHENTICATED; AND
STORES A KEY ASSOCIATED WITH THE APPLICATION CONTAINER DATA STRUCTURE.

Fig. 9

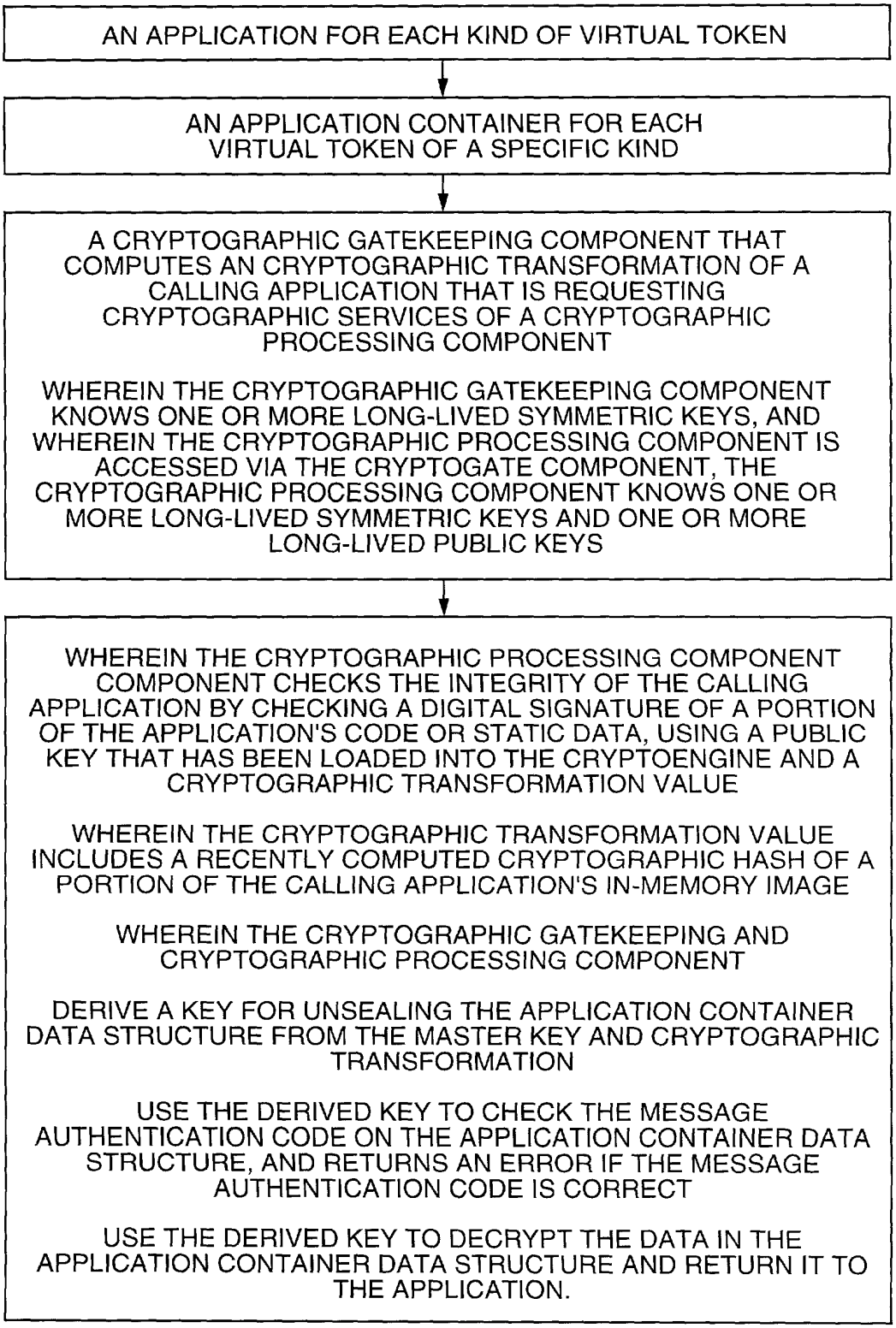


Table 1

AppContainer Structure

Offset	Size	Field Name	Description
0x00	1 bytes	OpCode	Indicates contents and format of the data field
0x01	1 bytes	Format	FmtAppContainer = 2
0x02	4 bytes	Reserved	0. This will be used in the future for extended opcode information.
0x06	2 bytes	Length	Count of bytes from the AppCodeDigest field up to and including the Data field. Count of bytes after seal operation but before ciphertext replacement. Count includes fields from ACD up to and including the Pad field.
0x08	20 bytes	AppCodeDigest (ACD)	Result of the SHA-1 digest of owning code that has been encrypted by the Enc160Bits primitive.
0x1c	16 bytes	InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode. IV is passed in by the OSD Security module.
0x2c	20 bytes	SealerCodeDigest (SCD)	Result of SHA1 digest of code for the program that sealed this container. Normally SCD is equal to the ACD. The SCD is set to zero if the container was sealed by the Device Authority server. It could also be the digest of another program if the program was authorized to transfer containers to this one. The SCD is passed in by the OSD Security module.
0x40	0-4096 bytes	Data	Data with a format determined by the OpCode
Varies	20 bytes	MAC	HMAC cryptographic primitive = HMAC (NewKey(Key,UsageAppMac), Payload)
Varies	1-16[1] bytes	Pad	Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer.

Table 2

Structure Modifications during OSD AppContainer Sealing

Field Name	OSD Sealing Phase before sending to SMI Layer
OpCode	Indicates contents and format of the data field
Format	FmtAppContainer = 2
Reserved	0. This will be used in the future for extended opcode information.
Length	Count of bytes from the AppCodeDigest field up to and including the Data field.
AppCodeDigest (ACD)	Result of the SHA1 digest of owning code that has been encrypted by the Enc160Bits primitive.
InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode.
SealersCodeDigest (SCD)	Result of SHA1 digest of code for the program that sealed this container. Normally SCD is equal to the ACD. It could also be the digest of another program if the program was authorized to transfer containers to this one.
Data	Data with a format determined by the OpCode.
MAC	NULL
Pad	NULL

Table 3

Structure Modifications during SMI AppContainer Sealing

Field Name	SMI Sealing Phase I
OpCode	Indicates contents and format of the data field
Format	FmtAppContainer = 2
Reserved 0.	This will be used in the future for extended opcode information.
Length	Count of bytes after seal operation but before ciphertext replacement. Count includes fields from ACD up to and including the Pad field.
AppCodeDigest (ACD)	Result of the SHA1 digest of owning code that has been encrypted by the Enc160Bits primitive.
InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode
SealersCodeDigest (SCD)	Result of SHA1 digest of code for the program that sealed this container. Normally SCD is equal to the ACD. It could also be the digest of another program if the program was authorized to transfer containers to this one.
Data	Data with a format determined by the OpCode.
MAC	HMAC cryptographic primitive= HMAC NewKey(Key,UsageAppMac), Payload)
Pad	Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer

Table 4

Final Sealed Structure

Field Name	SMI Sealing Final
OpCode	Indicates contents and format of the data field
Format	FmtAppContainer = 2
Reserved	0. This will be used in the future for extended opcode information.
Length	Count of bytes after seal operation but before ciphertext replacement. Count includes fields from ACD up to and including the Pad field.
AppCodeDigest (ACD)	Result of the SHA1 digest of owning code that has been encrypted by the Enc160Bits primitive.
InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode
SealersCodeDigest (SCD)	Result of SHA1 digest of code for the program that sealed this container. Normally SCD is equal to the ACD. It could also be the digest of another program if the program was authorized to transfer containers to this one.
Data	Data with a format determined by the OpCode.
MAC	HMAC cryptographic primitive= HMAC(NewKey(Key,UsageAppMac, Payload)
Pad	Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer.

Table 5

MKContainer Structure

Offset	Size	Field Name	Description
0x00	1 bytes	OpCode	Indicates contents and format of the data field
0x01	1 bytes	Format	FmtMkContainer
0x02	4 bytes	Reserved	0. This will be used in the future for extended opcode information.
0x06	2 bytes	Length	Count of remaining bytes with MSB first. For a sealed container this includes the length of the Mac and Padding bytes, for an unsealed container it does not include either the Mac or Padding byte lengths (i.e., it specifies the total byte length of items MKDigest through Data).
0x08	20 bytes	MKDigest	20 byte result of SHA1 digest of the master key.
0x1c	16 bytes	InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode. IV is passed in by the OSD Security module.
0x2c	20 bytes	SealersCodeDigest (SCD)	Result of SHA1 digest of code for the program that sealed this container. The SCD is set to zero if the container was sealed by the Device Authority server. The SCD is passed in by the OSD Security module.
0x40	0-64000 bytes	Data	Data with a format determined by the OpCode.
Varies	20 bytes	MAC	HMAC cryptographic primitive = HMAC (NewKey(Key, UsageMKMac), Payload)
Varies	1-16 bytes		Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer

Table 6

Structure Modifications during OSD MKContainer Sealing

Field Name	OSD Sealing Phase before sending to SMI Layer
OpCode	Indicates contents and format of the data field
Format	FmtAppContainer
Reserved	0. This will be used in the future for extended opcode information.
Length	Count of bytes after seal operation but before ciphertext replacement. Count includes fields from MKDigest up to and including the Pad field.
MKDigest	20 byte result of SHA1 digest of the master key.
InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode
SealersCodeDigest (SCD)	Result of SHA1 digest of code for the program that sealed this container
Data	Data with a format determined by the OpCode.
MAC	HMAC cryptographic primitive= HMAC(NewKey(Key,UsageAppMac, Payload)
Pad	Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer.

Table 7

Final Sealed Structure

Field Name	SMI Sealing Final
OpCode	Indicates contents and format of the data field
Format	FmtMKContainer
Reserved	0. This will be used in the future for extended opcode information.
Length	Count of bytes after seal operation but before ciphertext replacement. Count includes fields from MKDigest up to and including the Pad field.
MKDigest	20 byte result of SHA1 digest of the master key.
InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode
SealersCodeDigest (SCD)	Result of SHA1 digest of code for the program that sealed this container
Data	Data with a format determined by the OpCode.
MAC	HMAC cryptographic primitive= HMAC(NewKey(Key,UsageAppMac, Payload)
Pad	Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer.

Table 8

SignedContainer Structure

Offset	Size	Field Name	Description
0x00	1 bytes	OpCode	Indicates contents and format of the data field
0x01	1 bytes	Format	FmtMkContainer
0x02	4 bytes	Reserved	0. This will be used in the future for extended opcode information.
0x06	2 bytes	Length	Count of remaining bytes with MSB first. For a sealed container this includes the length of the Mac and Padding bytes, for an unsealed container it does not include either the Mac or Padding byte lengths (i.e., it specifies the total byte length of items MKDigest through Data).
0x08	20 bytes	PublicKeyDigest	SHA1 digest of the public key that should be used to verify the signature block.
			Random initialization vector for Cipher Block Chaining (CBC) mode. IV is passed in by the OSD Security module.
0x28	0-64000 bytes	Data	Data with a format determined by the OpCode.
Varies	128 bytes	SigRSABlock	When unsealed, this field begins with padding bytes set to zero and ends with a 20-byte Digest value. The Digest is the SHA1 digest of Opcode || Format || Unsealed-Length || PublicKeyDigest || Data. The sealed version of this field is RSA encrypted with a private key

Table 9

Final Sealed Structure

Field Name	Description
OpCode	Indicates contents and format of the data field
Format	FmtSignedContainer
Reserved	0. This will be used in the future for extended opcode information.
Length	Count of remaining bytes with MSB first. The unsealed length includes the PublicKeyDigest and the Data but not the SigRSABlock. The sealed length include the 128 bytes of the SigRSABlock.
PublicKeyDigest	SHA1 digest of the public key that should be used to verify the signature block.
Data	Data with a format determined by the OpCode.
SigRSABlock	When unsealed, this field begins with padding bytes set to zero and ends with a 20-byte Digest value. The Digest is the SHA1 digest of Opcode || Format || Unsealed-Length || PublicKeyDigest || Data. The sealed version of this field is RSA encrypted with a private key

Table 10

PubKcontiner structure with embedded MKContainer

Offset	Size	Field Name	Description
0x00	1 bytes	OpCode	Indicates contents and format of the data field
0x01	1 bytes	Format	FmtPubKContainer
0x02	4 bytes	Reserved	0. This will be used in the future for extended opcode information.
0x06	2 bytes	Length	Count of remaining bytes with MSB first. For a sealed container this includes the length of Mac and Padding bytes, for an unsealed container it does not include either the Mac or Padding byte lengths (i.e., it specifies the total byte length of items at offsets ###todo: get offsets).
0x08	20 bytes	PublicKeyDigest	Result of SHA1 digest of the public key (generally the Server Communication Key).
0x1c	128 bytes	PubKRSABlock	When unsealed this field begins with padding bytes set to zero and ends with Opcode || Format || KID || MK. These fields have fixed lengths. When sealed, this is an RSA encrypted value. The Opcode is item 1 above, not the Opcode for the MKContainer. If the first part is reused, the Opcode in the PubKRSABlock may not match item 1 but instead may be one of a small number of acceptable alternative values that indicate the reuse of the block.
Embeded MKContainer starts at offset 0x98			
+0x00	1 bytes	OpCode	Indicates contents and format of the data field
+0x01	1 bytes	Format	FmtMKContainer
+0x02	4 bytes	Reserved	0. This will be used in the future for extended opcode information.
+0x06	2 bytes	Length	Count of remaining bytes with MSB first. The unseal length includes items at offsets +0x04 to +0x3C, whereas the sealed length includes items at offsets.
+0x08	20 bytes	MKDigest	20 byte result of SHA1 digest of the Master Key stored in the 1st part PubKRSABlock.
+0x1c	16 bytes	Initialization Vector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode. IV is passed in by the OSD Security module.
+0x2c	20 bytes	SealersCode Digest (SCD)	Result of SHA1 digest of code for the program that sealed this container. The SCD is set to zero if the container was sealed by the Device Authority server. The SCD is passed in by the OSD Security module.
+0x40	0-64000 bytes	Data	Data with a format determined by the OpCode.
Varies	20 bytes	MAC	HMAC cryptographic primitive = HMAC (NewKey(Key, UsageMKMac), Payload)
Varies	1-16 bytes		Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer.

Table 11

Final Sealed PubKContainer Structure

Field Name	Description
OpCode	Indicates contents and format of the data field
Format	FmtPubKContainer
Reserved	0. This will be used in the future for extended opcode information.
Length	Count of remaining bytes with MSB first. For a sealed container this includes the length of the Mac and Padding bytes, for an unsealed container it does not include either the Mac or Padding byte lengths (i.e., it specifies the total byte length of items at offsets ###todo: get offsets).
PublicKeyDigest	Result of SHA1 digest of the public key (generally the Server Communication Key).
PubKRSABlock	When unsealed this field begins with padding bytes set to zero and ends with Opcode || Format || KID || MK. These fields have fixed lengths. When sealed, this is an RSA encrypted value. The Opcode is item 1 above, not the Opcode for the MKContainer. If the first part is reused, the Opcode in the PubKRSABlock may not match item 1 but instead may be one of a small number of acceptable alternative values that indicate the reuse of the block.
Embedded MKcontainer starts at offset 0x98	
OpCode	Indicates contents and format of the data field
Format	FmtMKContainer
Reserved	0. This will be used in the future for extended opcode information.
Length	Count of remaining bytes with MSB first. The unseal length includes items at offsets +0x04 to +0x3C, whereas the sealed length includes items at offsets.
MKDigest	20 byte result of SHA1 digest of the Master Key stored in the 1st part PubKRSABlock.
InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode. IV is passed in by the OSD Security module.
SealersCodeDigest (SCD)	Result of SHA1 digest of code for the program that sealed this container. The SCD is set to zero if the container was sealed by the Device Authority server. The SCD is passed in by the OSD Security module.
Data	Data with a format determined by the OpCode.
MAC	HMAC cryptographic primitive = HMAC (NewKey(Key, UsageMKMac), Payload)
Pad	Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer.

Table 12

Final Sealed PubKContainer Structure

Field Name	Description
OpCode	Indicates contents and format of the data field
Format	FmtPubKContainer
Reserved	0. This will be used in the future for extended information.opcode
Length	Count of remaining bytes with MSB first. For a sealed container this includes the length of the Mac and Padding bytes, for an unsealed container it does not include either the Mac or Padding byte lengths (i.e., it specifies the total byte length of items at offsets ###todo: get offsets).
PublicKeyDigest	Result of SHA1 digest of the public key (generally the Server Communication Key).
PubKRSABlock	When unsealed this field begins with padding bytes set to zero and ends with Opcode II Format II KID II MK. These fields have fixed lengths. When sealed, this is an RSA encrypted value. The Opcode is item 1 above, not the Opcode for the MKContainer. If the first part is reused, the Opcode in the PubKRSABlock may not match item 1 but instead may be one of a small number of acceptable alternative values that indicate the reuse of the block.
Embeded MKContainer starts at offset 0x98	
OpCode	Indicates contents and format of the data field
Format	FmtMKContainer
Reserved	0. This will be used in the future for extended opcode information.
Length	Count of remaining bytes with MSB first. The unseal length includes items at offsets +0x04 to +0x3C, whereas the sealed length includes items at offsets.
MKDigest	20 byte result of SHA1 digest of the Master Key stored in the 1st part PubKRSABlock.
InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode. IV is passed in by the OSD Security module.
SealersCode Digest (SCD)	Result of SHA1 digest of code for the program that sealed this container. The SCD is set to zero if the container was sealed by the Device Authority server. The SCD is passed by the OSD Security module.
Data	Data with a format determined by the OpCode.
MAC	HMAC cryptographic primitive = HMAC (NewKey(Key, UsageMKMac), Payload)
Pad	Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer.

Table 13

Final Sealed Structure

Field Name	SMI Sealing Final
OpCode	Indicates contents and format of the data field
Format	FmtMKContainer
Reserved	0. This will be used in the future for extended opcode information.
Length	Count of bytes after seal operation but before ciphertext replacement. Count includes fields from MKDigest up to and including the Pad field.
MKDigest	20 byte result of SHA1 digest of the master key.
InitializationVector (IV)	Random initialization vector for Cipher Block Chaining (CBC) mode
SealersCodeDigest (SCD)	Result of SHA1 digest of code for the program that sealed this container
Data	Data with a format determined by the OpCode.
MAC	HMAC cryptographic primitive = HMAC(NewKey(Key,UsageAppMac, Payload)
Pad	Number of Pad bytes is set to make sure that the Plaintext is a multiple of 16 bytes. Each padding byte has a value equal to the number of padding bytes in the Pad buffer.